



## UWS Academic Portal

### Automated task scheduling for automotive industry

Lewandowski, R.; Olszewska, J. I.

*Published in:*  
2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)

*DOI:*  
[10.1109/INES49302.2020.9147169](https://doi.org/10.1109/INES49302.2020.9147169)

Published: 27/07/2020

*Document Version*  
Peer reviewed version

[Link to publication on the UWS Academic Portal](#)

*Citation for published version (APA):*  
Lewandowski, R., & Olszewska, J. I. (2020). Automated task scheduling for automotive industry. In *2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)* (pp. 000159-000164). (IEEE Conference Proceedings). IEEE. <https://doi.org/10.1109/INES49302.2020.9147169>

#### General rights

Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

#### Take down policy

If you believe that this document breaches copyright please contact [pure@uws.ac.uk](mailto:pure@uws.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



## UWS Academic Portal

### Automated task scheduling for automotive industry

Lewandowski, R.; Olszewska, J. I.

*Published in:*  
IEEE International Conference on Intelligent Engineering Systems

*DOI:*  
[10.1109/INES49302.2020.9147169](https://doi.org/10.1109/INES49302.2020.9147169)

Published: 27/07/2020

*Document Version*  
Peer reviewed version

[Link to publication on the UWS Academic Portal](#)

*Citation for published version (APA):*  
Lewandowski, R., & Olszewska, J. I. (2020). Automated task scheduling for automotive industry. In *IEEE International Conference on Intelligent Engineering Systems* (pp. 000159-000164). (IEEE Conference Proceedings). IEEE. <https://doi.org/10.1109/INES49302.2020.9147169>

#### General rights

Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

#### Take down policy

If you believe that this document breaches copyright please contact [pure@uws.ac.uk](mailto:pure@uws.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Automated Task Scheduling for Automotive Industry

R. Lewandowski and J. I. Olszewska

University of West Scotland, United Kingdom

**Abstract**—Nowadays, the automotive industry requires an increased use of intelligent systems to endure. In this paper, we present a new solution for automated task scheduling to help automotive industry in efficiently managing garage employees' time and improving the effectiveness of the servicing and maintenance tasks of vehicles. The developed approach consists in a set of interconnected web applications with a model-view-controller based-on architecture and expert-knowledge temporal logic rules to automate the assignment of the daily workload for engineers working on multiple workstations within an automotive company. The proposed intelligent system prioritises and selects tasks for these engineers; the scheduled tasks being automatically ordered and displayed accordingly on screens visible in the related garage workstations. This automated task scheduling system has been successfully deployed within Arnold Clark Automobiles Ltd Company, and the performance of this new application used by a group of engineers under real-world operational conditions have been assessed and analysed.

**Index Terms**—Intelligent Systems, Industry 4.0 and Smart factory, Manufacturing-oriented Data Analytics, Intelligent Human-Machine Interaction, Expert Systems.

## I. INTRODUCTION

The introduction of Information Technologies (IT) in automotive industry is a new trend, which is fuelled by the development of both the Digitalization [1] and the Industry 4.0 [2], in order to cope with the major challenges the automotive industry is currently facing, such as the (a) complexity and cost pressure, (b) diverging markets, (c) digital demands, and (d) shifting industry landscape [3].

Indeed, the ongoing increase in regulations with respect to environmental and safety standards raises not only the production costs but also the manufacturing complexity, as it requires more platform sharing and more modular systems, leading e.g. to intelligent production in decentralized manufacturing systems (DMS) [4], cyber-physical production systems [5], or cellular manufacturing [6]. Moreover, the automotive industry needs to efficiently address the consumers' demands of active safety, connectivity, and ease of use, as well as the suppliers' pressure to provide more value-added content per car and sustained maintenance quality [7].

Hence, information technologies and especially automated planning and scheduling [8], [9] can aid the automotive industry to meet these demands [10]. In particular, automated task *planning* defines *what* and *how* the tasks have to be done, while the automated task *scheduling* defines *when* and *who* will do them [11]. Furthermore, scheduling defines priority and/or arranges activities to meet certain requirements, constraints, or objectives [12]; time being the major constraint (i.e. activities should be scheduled so that jobs could be accomplished within the available time) and a limiting resource (i.e.

activities should be scheduled to utilize this limited resource in an optimum manner) [13].

Process planning and industrial scheduling have a central function in manufacturing industries which consists in a systematic determination of the methods by which a product is to be manufactured competitively and economically [12]. Recently, advanced planning systems (APS) have become an important tool for manufacturing and production companies that require a specific system to optimize production, logistic, material and human resources, etc. with the goal of improving the economy of the company and/or offering a good customer service [14].

For this purpose, computer-aided process planning (CAPP) software packages have been developed to automate the planning and scheduling of industrial activities by using data structures such as lists [15], queues [16], tensors [17], graphs [18], or trees [19] to represent the activities and applying deterministic [20] or stochastic approaches [21] to allocate tasks, while minimizing the constraints with heuristics [22], optimization processes [23], tabu search [24], dispatching rules [25], auction mechanism [26], machine learning [27], or multi-agent systems (MAS) [28].

Current efforts of automated scheduling in the automotive industry are mostly focused on developing intelligent systems for enterprise resource planning (ERP) [10], automated production systems (aPS) [15], supply chain management (SCM) [29], including order-to-delivery (OTD) process [30], Sales and operations planning (S&OP) [31], warehouse system automation [32] and just-in-time routing for delivery [33].

However, little has been developed for the automation of repair activities and/or garage management which is still performed by humans without bespoke mobile open IT tools. Usual techniques involve manual task completion [34], Gantt chart [35], or Excel spreadsheet [36]. More advanced tools include Microsoft Elva DMS Scheduler [37], which is a planning tool for the management of the arrival time of cars, the completion of work of individual mechanics, and the car service operations, e.g. to coordinate work in both parts and service departments. One of the major disadvantages of this system is the lack of support for mobile devices. Another application which is currently available on the market is an integrated calendar and task scheduler made by International Business Machines Corporation [38]. The purpose of this application is to implement a method that is based on automatic segregation of commands for the user. This app has a built-in calendar that helps the user in planning all the necessary tasks.

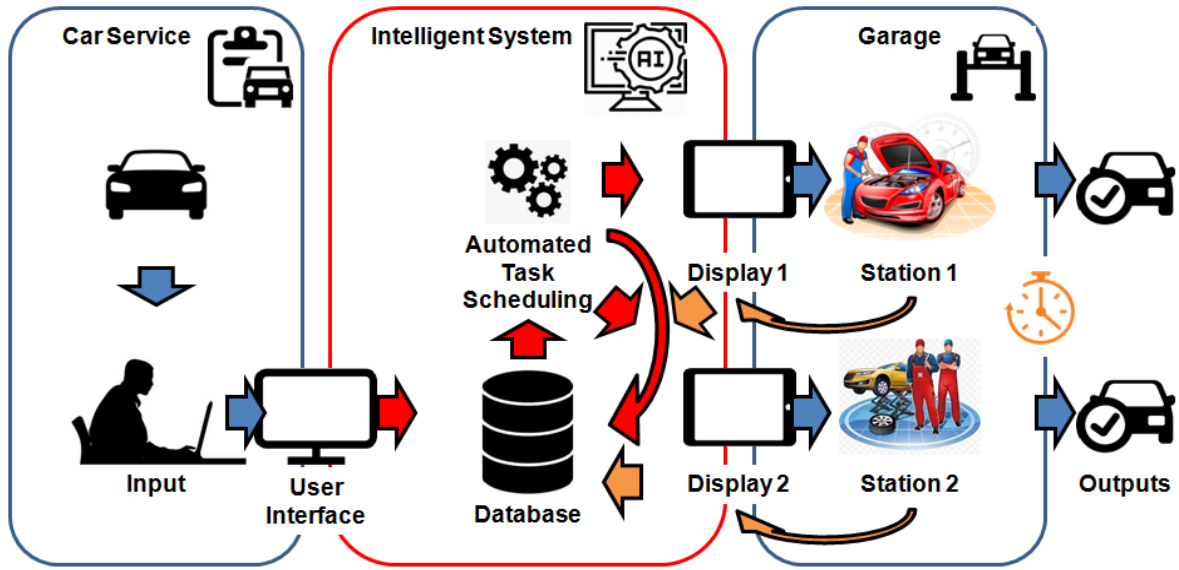


Fig. 1. Data flow within the automated scheduling system for automotive servicing and maintenance tasks.

The program orders all events in the calendar according to their validity. Tasks are organized into blocks that move according to the assumed algorithms. After the user has entered a new task into the application, the program identifies the task in a group of unscheduled and incomplete tasks which have the highest priority level in the group. Then, the application locates the next available block of free time in the user's calendar. Each new element or task in the calendar is arranged one after the other, and each new task becomes less important than the previous one. Consequently, the oldest task in the calendar is always the most important. This is a flaw in this approach, which is eliminated in our system by adding a priority factor for the tasks to be completed.

Hence, in this paper, we present a new automated scheduling system for managing the daily workload of engineers and garage station tasks (see Fig. 1). First, tasks (or jobs) are assigned an element of importance, which is given on a scale from 1 (highest priority) to 3 (lowest priority), based on the manager's expert knowledge. Then, the intelligent web application selects tasks for the engineers taking into account each task priority, related job makespan, workstations' availability, and engineers' time constraints. Next, the tasks are ordered and displayed accordingly on mobile screen dashboards which are visible in the garage workstations; the displayed tasks being automatically refreshed every 5s to accommodate the real-time flow of information.

Indeed, our intelligent system performs a *dynamic scheduling*, where real-time information is used as it arrives [39]. In particular, it performs a *predictive-reactive scheduling* [25], i.e. a scheduling/rescheduling process in which schedules are revised in response to real-time events. More specifically, the predictive-reactive scheduling is a two-step process, where a predictive schedule is generated in advance with the objective of optimising garage performance without considering possible disruptions/issues on the workstations, while this schedule could be modified during the execution in response to real-time events.

Besides, our automated task scheduling considers the *flexible job shop scheduling* paradigm [40], where  $n$  jobs  $J_n$ , which consist each in a set of operations  $O_{n,l}$ , are processed by engineers on the garage's shop floor with parallel machines (i.e.  $m$  workstations  $W_m$  which are equipped each with a pool of  $k$  identical machines with various constraints and costs), in a given job-dependent order.

It is worth noting that according to the expert knowledge, the time for completion of a given job on a particular workstation depends on a number of factors, typically including the complexity and duration of operations, the number and availability of the car's parts, and the engineers' skills. Thence, our system aims to assign a job to the given workstation not only based on the job's priority, but also in relation to the validity and timing of the task.

On the other hand, the developed web application is an automated system; the only interaction of the staff being the possibility of entering/changing tasks in the system database, as shown in Fig. 1.

The contribution of this paper is the automation of the scheduling task in context of garage service with multiple workstations, through the development of an innovative web-based, interconnected intelligent system and new temporal interval scheduling rules.

The paper is structured as follows. In Section II, we present our developed automated scheduler for car maintenance and service, while in Section III, we report and analyse the results of the testing and deployment of our web-based scheduling system. Conclusions are drawn up in Section IV.

## II. PROPOSED SCHEDULING SYSTEM

The proposed task scheduling system (Fig. 1) intends to aid garages to improve the quality of maintenance and reduce the time of servicing. Hence, the developed scheduler relies on temporal interval rules which are explained in Section II-A, while the implementation of the overall automated scheduling system is described in Section II-B.

### A. Time-Interval Scheduling

*Task Scheduling* is concerned with the allocation of resources over time to perform tasks. Hence, most of the existing task scheduling algorithms are checking the amount of time needed to perform a given activity and compare it to the free time which is allocated to complete the task. If the free block of time is not enough to finish any task, the program grants the second block of time to be able to fulfill the specified condition [41].

In particular, *Interval Scheduling* consists in the allocation of a time interval that is needed to complete a task. This approach has the aim of identifying the set of intervals which are not overlapping for the longest possible time. The goal of algorithms based on this approach is to find as many intervals as possible at a given time, one after another. One of such algorithms is the Least Earliest Completion Time First (LECF) algorithm [42], which computes a schedule in which the number of jobs that are completed in one of their feasible intervals is maximized. Indeed, time-critical jobs in many real-time applications have multiple feasible intervals, while each job is constrained to execute from start to completion in one of its feasible intervals. Thence, LECF is characterized by executing the job which is able to complete the fastest among the unselected jobs in a good manner. The research shows that LECF algorithm is able to execute more intervals in less time than previous heuristic algorithms. However, this method assumes that a job fails if the job remains incomplete at the end of the last feasible interval.

In this work, we adopt the interval scheduling approach we model with temporal interval logic rules. In particular, we model a time interval as an ordered set of points  $T = \{t\}$  defined by end-points  $t^-$  and  $t^+$ , leading to the formal logic definition [43] as follows:

$$(t^-, t^+) : (\forall t \in T)(t > t^-) \wedge (t < t^+). \quad (1)$$

Consequently, we define the makespan  $S$  of a job  $J$  as  $(t_s^-, t_s^+) : (\forall t_s \in S)(t_s > t_s^-) \wedge (t_s < t_s^+)$  and a feasible time interval  $F$  as  $(t_f^-, t_f^+) : (\forall t_f \in F)(t_f > t_f^-) \wedge (t_f < t_f^+)$ .

Based on the expert knowledge gathered by surveying both manager and mechanic staff, it appears that the makespan of each job lasts between 30 and 60 minutes. Hence, unlike [44] which considers that each job has a different processing time, our initial scheduling follows a robust approach which allocates 1-hour time intervals for each job on the active list to be completed. We assume also that a job does not fail if the job remains incomplete at the end of the last feasible interval, but instead this job is added to a waiting list to be rescheduled.

Further assumptions for our task scheduling system are as follows: (i) jobs ( $J_n$ ) and workstations ( $W_m$ ) are independent; (ii) each workstation can process at most one job at any time; (iii) each job requires only one single workstation to be processed; (iv) the set of machines ( $M_k$ ) in each workstation is identical; (v) each job is a set of specific operations  $O_i$ ; (vi) jobs are assigned a priority  $P_p \in [1, 2, 3]$ , with 1 the highest priority and 3 the lowest priority; (vii) ready times of all jobs are zero, i.e. all jobs are available at the commencement of the daily processing (for the initial scheduling); (viii) all

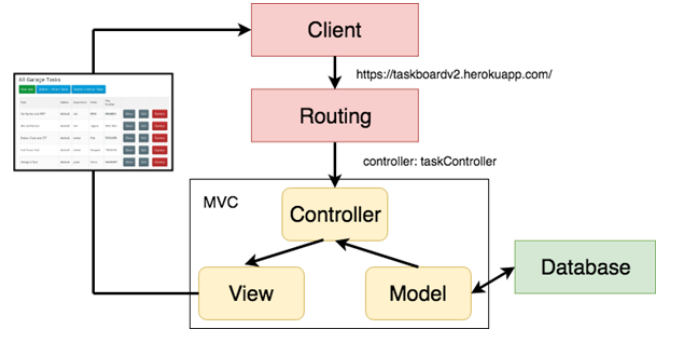


Fig. 2. Architecture of the intelligent system for automated task scheduling.

workstations are available at time zero (i.e. at the beginning of each day).

Our system first prioritises and selects tasks from the active list. The scheduler deals then with each new task from the active list in the way that each job in the active list must meet the given conditions, among others, that the length of the job must be shorter than the length of the block of time allocated at that time. However, if the allocated block of time is too short for the task, the system assigns the second block of time that fits for completing the task.

That car maintenance and service domain knowledge is used in this work to generate an initial rule to allocate a job  $J$  with its corresponding makespan  $S$  to a feasible interval  $F$  in whatever workstation  $W$ , which is formalized in temporal Descriptive Logic (DL) and leads to the axiom as follows:

$$\begin{aligned} assign(S@t_s, F@t_f) \sqsubseteq Scheduling\_Rule \\ \sqcap (\diamond t_s)(\diamond t_f) \\ (t_{s-} \geq t_{f-}) \sqcap (t_{s+} \leq t_{f+}) \\ \cdot ((S@t_s \sqcap F@t_f), \end{aligned} \quad (2)$$

where the temporal DL symbol  $\diamond$  represents the temporal existential qualifier.

### B. Implementation

The presented intelligent scheduling system (Fig. 1) has been developed following the Rapid Application Development (RAD) approach [45] and consists of two main applications. The first application has a customized Model-View-Controller (MVC) architecture (Fig. 2) and is responsible for the possibility of inputting tasks into the database (Fig. 3) and transferring them to the automated scheduler, as explained in Section II-B1. The second application's main responsibility is the presentation of these tasks in a proper order on the screen of the corresponding workstation dashboard (Fig. 4), as presented in Section II-B2.

1) *Task Data Entry Application*: The first application provides the data entry system (Fig. 2) and the dashboard with the tasks which are populated from the database application (Fig. 3). The task data entry application for the service department employees allows the garage manager to add new tasks for all engineers and stations.

This web application is stored on a central server. Users can access the system on any computer with internet access and

can receive the full functionality of the application by entering the corresponding web address in the browser.

The architecture of the application follows an MVC pattern [46] on which the Ruby on Rails structure is based and which defines three separate components as follows:

- **Model:** this component is responsible for connecting to the database and its relational-object mapping. It implements the business logic of the application. Models significantly simplify typical operations, such as data management, pagination, and independent application from more specifically, the version of the database. The model is responsible for combining data from the database and sorting it in the right way, i.e. sorting tasks for individual stations as well as sorting tasks by importance, where the most important task is of priority 1.
- **View:** it is a presentation layer of data provided by the model via the controller. The view consists of Embedded Ruby (ERB) templates containing, for the most part, HTML and JavaScript code with Ruby language additions. They are created dynamically by the application based on data provided by the controller.
- **Controller:** it is an intermediary component between the layers of model and view. Depending on the customer's request, the relevant data is downloaded from the model and then is made available to the view. Indeed, sorted data are called in the controller by their method. The completed response, in the form of a generated view, is then sent back to the client who made the request.

A simple schematic of the relationship between the modules of our architecture is shown in Fig. 2. It is worth noting that there is also an additional element in our architecture that does not belong to the MVC pattern, but closely related to it. That is the *routing* module. Based on the URL received, it decides which controller it applies to and the request is forwarded to it.

On the other hand, the application adopts the *Active Record* mechanism [46] which controls the process between our database and the application. In the Active Record scheme, the user can perform operations on database tables using record objects. Each record represents a row in the database table, and each Active Record object has a set of four methods, namely, Create, Read, Update, Delete (CRUD) for accessing the database. Hence, after adding a new task to the database, information about the added task appears on the screen, which can be edited if necessary (Fig. 3).

After completing the addition of a new task, the task is placed on the task list, and the user can see where the program assigned that task. If the task has a low priority, it is assigned to a waiting list. If the task has a high priority, it is assigned to the active list of the given station and is displayed on the screen in the garage with the time which has been assigned by the dashboard application.

The data entry application uses a PostgreSQL relational database to store all the tasks and the Heroku cloud platform. This program acts like a Restful API and provides the JavaScript Object Notation (JSON) file with all the data needed to populate the dashboard application, which is visible for engineers in the garage; the method to transfer data between

## All Garage Tasks

Task	Station	Importance	Make	Reg Number			
Car Service and MOT	station1	red	BMW	BM66BIG	Show	Edit	Destroy
Mot and Service	station1	red	Jaguar	WN110LH	Show	Edit	Destroy
Brakes Check and CIT	station1	amber	Fiat	FD54WER	Show	Edit	Destroy
Fuel Pump Fault	station1	amber	Peugeot	TBE69LOL	Show	Edit	Destroy
Change a Tyre	station1	green	Volvo	GG65WRT	Show	Edit	Destroy

Fig. 3. Overview of the manager's task scheduling application.

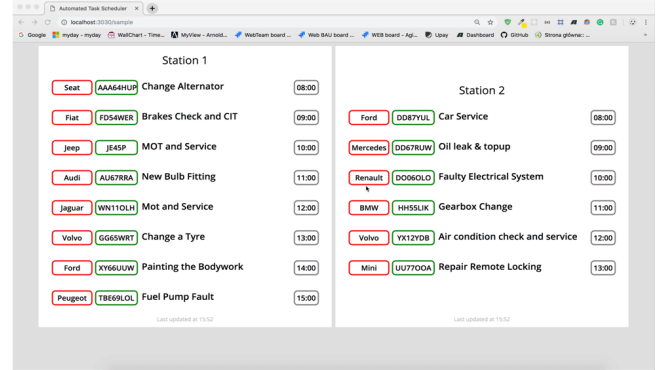


Fig. 4. Automated scheduling results displayed on the task dashboards of the workstations.

the two applications being the JSON response built with the JBuilder gem.

2) *Task Scheduling Dashboard Application:* The second application displays the tasks on the screen of each workstation dashboard (Fig. 4). We can notice that the size of each window has been set evenly using `widget_base_dimension = [70,70]`.

The allocation of time for each task is done at the level of the dashboard application model. After retrieving data from the JSON file generated by the first application by connecting via the HTTP client with the provided URL link to download the content of the JSON file, time intervals are assigned to each task for each station separately.

The structure of data used in the task scheduling dashboard application are priority queues that were implemented for the purpose of the tasks that need to be prioritized in terms of their importance. Priority queues use a push-and-pop system that allows the system to quickly manage tasks which were sent from the database.

The application automatically refreshes the page with displayed tasks, using the Rufus scheduler gem which is an open-source Ruby on Rails library; the refreshing time between the updates of the tasks on the board being set to 5s.

## III. VALIDATION AND DISCUSSION

Our proposed intelligent system has been thoroughly tested and deployed under operating conditions of an automotive service department.



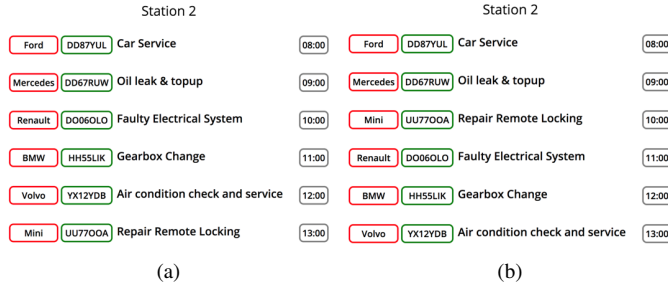


Fig. 5. Results of the (a) scheduling and (b) rescheduling displayed on the task dashboard of the workstation 2.

Prior to the real-world deployment of our system, unit tests were run first using RSpec which is a gem (library) that enabled the tests of the application elements using the terminal in the Ruby on Rails environment. Then, integration tests were carried out to check the interaction between multiple controllers and their actions to ensure that they work together in the expected manner. The syntax resembles functional tests, enriching them with an additional set of assertions coming from the Ruby on Rails ActionController::IntegrationTest class, which is the base class for all integration tests. Next, functional testing have been performed to test the scheduling and rescheduling functionalities of our system, as exemplified in Fig. 5.

Once the system was deployed, usability tests were conducted on a pilot group of fourteen people (i.e. 11 males and 3 females, aged between 18 and 65) which completed an online survey. According to the survey results, the users did not experienced any problem while using the system, and they found the system simple to learn and easy to use. Moreover, they all agreed that the application helped them to do more tasks during their shift.

Further experiments have been carried out to assess the performance of the intelligent system when deployed in its real-world environment. These experiments consisted in scheduling tasks of a garage which comprised two workstations and two engineer teams. Thence, tests were performed in this garage during one week without the automated system and then the week after, with the automated scheduling system. The overall experiment was repeated twice over a period of two months in the same garage.

Table I presents the results of these experiments in terms of the average number of finished tasks in the two workstations. We can observe that the proposed automated task scheduling system improves the efficiency of the workstations and engineers' time management by 9.3% compared to the manual system of planning and assigning tasks to employees. Indeed, without the automated system, mechanics must go to the service department office to receive a new task. Thus, employees lose a lot of time getting to know the new task and collecting the keys of the cars. With our intelligent system, the corresponding engineers' time is managed more efficiently, and the service department is able to service more cars within a week. The test data show that the number of tasks performed during the week with the system is on average raised by 3 tasks, and thus the customers' waiting time is decreased.

TABLE I  
AVERAGE NUMBER OF FINISHED TASKS IN AUTOMATIVE STATIONS.

Day of the Week	Without System	With Our System
Monday	6.25±0.83	<b>7.50±1.50</b>
Tuesday	8.00±1.00	<b>8.50±0.50</b>
Wednesday	7.25±1.30	<b>8.00±1.22</b>
Thursday	6.00±0.71	<b>7.25±0.43</b>
Friday	<b>7.50±1.11</b>	7.00±1.58
<b>Total</b>	35.00±0.71	<b>38.25±1.30</b>

On the other hand, costs have been decreased when using our automated system. In particular, the average estimated revenue in the service department per task is £212, which gave a weekly increase in the turnover of each station during the system testing period of £636 more than the weekly revenue without the use of an automatic system. On a monthly scale, the financial gain is estimated at £2,544 per station. It is worth noting that these results are very beneficial for the company, because the average cost of the system installation is around £500 per station. That means that the investment in the new automated scheduling system is paid back in the first week of the system operation.

Besides, the environmental cost of doing everything on paper is high, and the use of task dashboard intelligent application could contribute to lower company's carbon footprint.

#### IV. CONCLUSIONS

As the automotive industry tends to rely more and more on intelligent manufacturing systems and advanced planning systems, we presented in this paper an innovative scheduling system to efficiently handle daily jobs of workstations which are dedicated to car's maintenance and service. Our system is a working set of interconnected applications that dynamically react to existing events. In particular, it consists of two main web applications: one for database management of new or existing tasks and one for scheduling and displaying the assigned tasks to the corresponding workstations in the appropriate order and within the scheduled time intervals. Our system was successfully deployed in real-world and real-time operating conditions of an automotive service department. The tests demonstrated the intelligent scheduling system streamlined the department, speeded up its work, saved money for the automotive company, and made a positive environmental impact.

#### ACKNOWLEDGMENT

The authors would like to thank Arnold Clark Automobiles Ltd for the support of this work.

#### REFERENCES

- [1] S. Leonard, I. K. Allison, and J. I. Olszewska, "Design and test (D&T) of an in-flight entertainment system with camera modification," in *Proceedings of the IEEE International Conference on Intelligent Engineering Systems*, 2017, pp. 151–156.
- [2] V. R. Sampath Kumar, A. Khamis, S. Fiorini, J. L. Carbonera, A. Olivares Alarcos, M. Habib, P. Goncalves, H. Li, and J. I. Olszewska, "Ontologies for Industry 4.0," *Knowledge Engineering Review*, vol. 34, pp. 1–14, 2019.



- [3] D. Mohr, N. Muller, A. Krieg, P. Gao, H.-W. Kaas, A. Krieger, and R. Hensley, "The road to 2020 and beyond: What's driving the global automotive industry?," <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-road-to-2020-and-beyond-whats-driving-the-global-automotive-industry>, 2020.
- [4] L. Bochmann, T. Baenziger, A. Kunz, and K. Wegener, "Human-robot collaboration in decentralized manufacturing systems: An approach for simulation-based evaluation of future intelligent production," *Procedia CIRP*, vol. 62, pp. 624–629, 2017.
- [5] S. Mayer, C. Arnet, D. Gankin, and C. Endisch, "Standardized framework for evaluating centralized and decentralized control systems in modular assembly systems," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 113–119.
- [6] A. Saboor, M. Imran, M. H. Agha, and W. Ahmed, "Flexible cell formation and scheduling of robotics coordinated dynamic cellular manufacturing system: A gateway to Industry 4.0," in *Proceedings of the IEEE International Conference on Robotics and Automation in Industry*, 2019, pp. 1–6.
- [7] A. Abolhassani, E. J. Harner, and M. Jaridi, "Empirical analysis of productivity enhancement strategies in the North American automotive industry," *International Journal of Production Economics*, vol. 208, pp. 140–159, 2019.
- [8] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*, Morgan Kaufmann Publishers, 2010.
- [9] D. R. Sule, *Production Planning and Industrial Scheduling - Examples, Case Studies and Applications*, CRC Press, 2nd edition, 2007.
- [10] T. Staebelin and K. Aoki, "Planning and scheduling in the automotive industry: A comparison of industrial practice at German and Japanese makers," *International Journal of Production Economics*, vol. 162, pp. 258–272, 2015.
- [11] D. S. Nau, M. Ghallab, and P. Traverso, *Automated Planning and Acting*, Cambridge University Press, 2016.
- [12] F. Azemi, G. Simunovic, R. Lujic, and D. Tokody, "Intelligent computer-aided resource planning and scheduling of machining operation," *Procedia Manufacturing*, vol. 32, pp. 331–338, 2019.
- [13] B. C. McRae, *Practical Time Management: How to Get More Things Done in Less Time*, Self-Counsel Press, 1992.
- [14] M. C. Vidoni and A. R. Vecchietti, "A systemic approach to define and characterize advanced planning systems (APS)," *Computers and Industrial Engineering*, vol. 90, pp. 326–338, 2015.
- [15] K. Land, S. Cha, and B. Vogel-Heuser, "An approach to efficient test scheduling for automated production systems," in *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN)*, 2019, pp. 449–454.
- [16] W. Weng and S. Fujimura, "Online scheduling of flexible flow shop manufacturing," in *Proceedings of the IEEE International Joint Conference on Computational Sciences and Optimization*, 2009, pp. 112–116.
- [17] M. Kumar, S. Teso, P. De Causmaecker, and L. De Raedt, "Automating personnel rostering by learning constraints using tensors," in *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, pp. 697–704.
- [18] W. Sun, Y. Zhu, Z. Su, D. Jiao, and M. Li, "A priority-based task scheduling algorithm in grid," in *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms and Programming*, 2010, pp. 311–315.
- [19] M. Lubosch, M. Kunath, and H. Winkler, "Industrial scheduling with Monte Carlo tree search and machine learning," *Procedia CIRP*, vol. 72, pp. 1283–1287, 2018.
- [20] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Springer, 5th edition, 2016.
- [21] J. Framinan and R. Ruiz, "Architecture of manufacturing scheduling systems: Literature review and an integrated proposal," *European Journal of Operational Research*, vol. 205, no. 2, pp. 237–246, 2010.
- [22] B. L. MacCarthy and J. Liu, "Addressing the gap in scheduling research: A review of optimization and heuristic methods in production scheduling," *International Journal of Production Research*, vol. 31, no. 1, pp. 59–79, 1993.
- [23] K. B. Gavali, A. K. Bewoor, and D. Barik, "Effective utilization of job shop scheduling in auto industries with the aid of social spider optimization," *Journal of Green Engineering*, vol. 8, no. 4, pp. 475–496, 2018.
- [24] S. Kawaguchi and Y. Fukuyama, "Reactive tabu search for job-shop scheduling problems," in *Proceedings of the IEEE International Conference on Computer Science and Education*, 2016, pp. 97–102.
- [25] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *Journal of Scheduling*, vol. 12, no. 4, pp. 417–431, 2009.
- [26] Y. Yao, T. Kaihara, K. Sashio, and S. Fujii, "A study on automated scheduling methodology for machining job shop," in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2007, pp. 1018–1023.
- [27] B. Waschneck, A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, and A. Kyek, "Optimization of global production scheduling with deep reinforcement learning," *Procedia CIRP*, vol. 72, pp. 1264–1269, 2018.
- [28] A. Rajabinasab and S. Mansour, "Dynamic flexible job shop scheduling with alternative process plans: An agent-based approach," *International Journal of Advanced Manufacturing Technology*, vol. 54, pp. 1091–1107, 2011.
- [29] C. H. Lim, S. K. Moon, and E. S. Okpoti, "A reusable scheduling problem decomposition framework for smart factories," in *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, 2019, pp. 516–520.
- [30] P. G. Brabazon and B. L. MacCarthy, "The automotive order-to-delivery process: How should it be configured for different markets?," *European Journal of Operational Research*, vol. 263, no. 1, pp. 142–157, 2017.
- [31] N. Lahloua, A. El Barkany, and A. El Khalfi, "Sales and operations planning (S&OP) concepts and models under constraints: Literature review," *International Journal of Engineering Research in Africa*, vol. 34, pp. 171–188, 2018.
- [32] L. Sabattini, V. Digani, M. Lucchi, C. Secchi, and C. Fantuzzi, "Mission assignment for multi-vehicle systems in industrial environments," *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 268–273, 2015.
- [33] K. Nishida, T. Nishi, H. Kaname, K. Kumagai, and T. Higashi, "Just-in-time routing and scheduling for multiple automated guided vehicles," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 841–846.
- [34] L. Yang, J. Li, P. Hackney, F. Chao, and M. Flanagan, "Manual task completion time estimation for job shop scheduling using a fuzzy inference system," in *Proceedings of the IEEE International Conference on Internet of Things*, 2017, pp. 139–146.
- [35] X. Liu, Z. Shen, and L. Dai, "Scheduling optimization of flexible flow shop," in *Proceedings of the IEEE Information Technology and Mechatronics Engineering Conference*, 2018, pp. 1652–1657.
- [36] J. C. de Man, T. Nehzati, E. Arica, and K. Kiil, "Automating scheduling tasks: Experience of a real case," in *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, 2015, pp. 1447–1451.
- [37] Microsoft, "Elva DMS. Scheduling and Planning for Automotive Industry," <http://www.elvadm.com/features/service-management/scheduling-planning/>, 2017.
- [38] C. J. Jonhson, J. E. Shewbert, P. L. Statham, S. L. Tipton, and K. K. Yellepeddy, "US Patent-8645182 B2. Integrated Calendar and Task Scheduler," <https://patents.google.com/patent/US8645182B2/en>, 2014.
- [39] P. Cowling and M. Johansson, "Using real time information for effective dynamic scheduling," *European Journal of Operational Research*, vol. 139, no. 2, pp. 230–244, 2002.
- [40] J. Xie, L. Gao, K. Peng, X. Li, and H. Li, "Review on flexible job shop scheduling," *IET Collaborative Intelligent Manufacturing*, vol. 1, no. 3, pp. 67–77, 2019.
- [41] S. S. Yau and Zhou X, "Schedulability in model-based software development for distributed real-time systems," in *Proceedings of the IEEE International Workshop on Object-Oriented Real-Time Dependable Systems*, 2002, pp. 45–52.
- [42] J.-J. Chen, J. Wu, C.-S. Shih, and T.-W. Kuo, "Approximation algorithms for scheduling multiple feasible interval jobs," in *Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2005, pp. 11–16.
- [43] J. I. Olszewska, "Temporal interval modeling for UML activity diagrams," in *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2016, pp. 199–203.
- [44] A. Dolgui, D. Ivanov, S. P. Sethi, and B. Sokolov, "Scheduling in production, supply chain and Industry 4.0 systems by optimal control: Fundamentals, state-of-the-art and applications," *International Journal of Production Research*, vol. 57, no. 2, pp. 411–432, 2019.
- [45] N. Amanquah and O. T. Eporwei, "Rapid application development for mobile terminals," in *Proceedings of the IEEE International Conference on Adaptive Science and Technology*, 2009, pp. 410–417.
- [46] M. Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2002.